University of Amsterdam

UvA-DARE (Digital Academic Repository)

A semantic model for complex computer networks : the network description language

van der Ham, J.J.

Publication date
2010

Link to publication

Citation for published version (APA):
van der Ham, J. J. (2010). *A semantic model for complex computer networks : the network description language*.

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:23 Sep 2021

Chapter 2

Describing Computer Networks

2.1 Introduction

In the previous chapter we have shown that a main problem for requesting lightpaths or managing complex networks is the lack of topology descriptions. In this chapter we provide an overview of the current approaches to describing computer networks. It is important to note that any description requires two kinds of models[34]: information models that describe resources at a conceptual layer, and data models that describe protocol and implementation details. Obviously there is a relation between data models and information models, as a data model implements an information model. A relation can also exist the other way; if a certain data model is preferred, then this can impose limitations on how the information model is expressed.

The rest of this chapter is organised as follows: in section 2.2 we first look at the requirements of network descriptions in light of our research question. With these requirements, we review existing information models for computer networks in section 2.3. We show that none of the existing information models provide a good solution to our problem. In section 2.4 we examine other approaches that are used in routing protocols.

Before moving onto chapter 3 where we introduce our own information model, we look at possible data models, and provide a brief introduction to Resource Description Framework and the Semantic Web in section 2.5. We summarize and conclude our overview of information and data models in section 2.6.

2.2 Requirements for a Network Model

In the previous chapter we have shown that there are several steps involved in getting a lightpath. The whole provisioning process goes from submitting a clear request to a network provider, distributing the request to other networks, to testing and delivering the lightpath. Currently automating the process of requesting and creating a lightpath is hampered by the lack of network topology descriptions.

A model for network topology descriptions for a global optical network must fulfil the following requirements:

Concise The network model must provide a clear description of the network topology, and all the nodes involved. The definitions of elements must be clear to all the actors involved.

Interoperable There are different vendors for control plane software, which are mostly incompatible. The network descriptions must be in such a form that they can easily be parsed by applications.

Distributed The global network consists of a large number of domains, each managing its resources. Each domain must be able to maintain their own descriptions, and information should stay with its owner. The descriptions should somehow link to each other to form a description of the global network.

Portable Network descriptions will be exchanged between different domains, which means that the model must make use of portable identifiers, which are suitable for use in a global context.

Extensible The model must allow for easy extension or combination with descriptions of other resources. The networks are only vehicles for transporting data. The true value of the network is through the services that are provided at the edges, such as compute services, instruments, or data storage.

Human Readable Currently the process of provisioning lightpaths is done mostly manually. Because of the many parties involved, there will be a transition period where some domains are still provisioned manually, while

Name	Data Model	Expressivity	Portability	Activity	Human Readable	Standards Body
SNMP	ASN.1	-	-	dormant	-	IETF
NetConf	XML	+	+	active	+	IETF
CIM	XML	++	+	active	+	DMTF
GMPLS	OSPF LSAs	+	-	active	-	IETF
NM-WG	XML	+	+	active	+	OGF
cNIS	SQL	+	-	active	+	–
G.805	-	++	-	low	+	ITU-T

Table 2.1: *Feature comparison of information models, for details and abbreviations see section 2.3.*

others are automated. The descriptions should be human readable or tools should be provided to easily visualize them.

2.3 Information Models

In this section we provide an overview of current information models that are in use for describing network topologies. Table 2.1 shows different information models, along with their data models. The table also scores different features of the information and data models. However, not all requirements given in the previous section are easily quantifiable, so we discuss each of the information models in more detail below.

SNMP The Simple Network Management Protocol[1][35] is a set of standards describing a protocol, a database schema, and data objects. The whole suite was originally created as a way of both monitoring and managing network resources. In current networks it is mostly used only for monitoring purposes. Diagnostic, performance and configuration information of network devices can be retrieved from the Management Information Base (MIB) of devices. The MIB is a tree

[1] Technically, the information model is formed by the MIBs, Management Information Bases, and SNMP denotes the whole set: protocol, information and data model.

of name, value pairs, which can be requested and changed. A large part of the MIB is standardised, but vendors also have their own part of the tree. This vendor space is used to store most configuration and performance data of their devices in a proprietary format. Almost all networking devices support SNMP, with different levels of detail in their MIB.

The network description provided by SNMP is distributed over the devices. Depending on the layer the device is operating on, it may have a pointer (address or identifier) to its neighbours on that layer. A view of the whole topology can be created by combining the information gathered from all the devices.

NetConf The Internet Engineering Task Force (IETF) is currently working to replace SNMP with a new standard, NetConf[36]. While SNMP uses its own protocol and only allows for three data-types (integer, string, sequence) , NetConf uses XML, allowing for any data type. Netconf defines a way of transporting monitoring data and change requests over standard protocols, SSH, BEEP or SOAP. NetConf is aimed at distributing diagnostic, performance and configuration information, but also managing devices. NetConf is currently being introduced in networking devices.

As NetConf follows similar principles as SNMP, the network description provided by NetConf is similarly distributed over the managed devices. Each device will have information about the neighbour it connects to on the layer it operates on. The network topology can be created by combining the information of the devices in the network.

CIM Another network device information model is the Common Information Model, CIM[37]. It is being developed by the Distributed Management Task Force (DMTF)[38] and it is an object-oriented information model described using the Unified Modelling Language. The model captures information regarding computer systems, operating systems, networks and other related diagnostic information. CIM is a very broad and complex model, the current UML schemata of the network model is almost 40 pages, the total model is over 200 pages.

DMTF has defined a mapping from CIM to XML, which is mainly used in Web-Based Enterprise Management (WBEM). WBEM is mainly implemented in enterprise-oriented computing equipment, and operating systems such as Windows and Solaris. The high expressiveness in CIM combined with the very active development is to us a negative property. There have been many significant changes in the infrastructure part of CIM over the past two years.

GMPLS GMPLS, Generalized Multi-Protocol Label Switching[39], is a protocol suite developed by the IETF for the provisioning and management of label-switched paths through multi-technology networks. It provides a unified control and management plane for the management of multi-layer networks. Networking devices use the Open Shortest Path First Traffic Engineering (OSPF-TE) protocol to exchange topology data with their neighbours. Devices broadcast the received topology data to their other neighbours, so that in the end all the devices in the domain have the same view of the network topology. The IETF is currently trying to extend GMPLS towards the inter-domain environment.

The topology data in OSPF-TE is exchanged in Link State Announcements (LSAs) packets. The topology data contained therein is encoded in a compact byte format, using specifically defined header fields and Type-Length-Value (TLV) containers. This format is aimed at being easy to process and store for participating network devices, but it is hard to export to external applications.

NM-WG The Network Measurements Working Group (NM-WG) of the OGF has also developed a way of representing network topologies[40, 41]. The NM-WG schema can describe network topologies as used in network measurements. They have created an XML schema, used in several network measurement and assessment tools. Descriptions in this format are for example used by the performance measurement tool PerfSONAR[42]. This tool can collect information and parse output from other tools, and expresses the resulting information in XML. This allows for easy exchange of topology data with other tools.

cNIS cNIS[43] has been developed in the GÉANT2[44] project as a way to store topology information of a single domain. This information is collected and stored by different components such as the perfSONAR measurement tool, the AutoBAHN[45] provisioning tool and others. The model is defined in UML and the tables of the database. The cNIS server can provide the data in different formats towards the different tools. cNIS has a very detailed information model allowing it to describe single domain topologies on different layers. This model has also been brought into the NML-WG. The current plans are that once the NML-WG produces a standard model cNIS will move towards a unified interface for all network tools using that model.

G.805 G.805 is an information model defined by the ITU-T to describe end-to-end connections through a multi-technology network. It is a model to describe the theoretical foundation of network technologies and relations between them.

As such, G.805 only has a graphical data model. We will come back to the G.805 model in section 3.5.

NML-WG In 2007 the Network Markup Language Working Group (NML-WG)[46] was formed in the OGF. This working group has taken the topology schema of NM-WG, the Network Description Language and other schemata as input to define a new schema specifically for describing multi-layer network topologies in a broader application area.

2.3.1 Comparing Information Models

If we examine the requirements that we stated in section 2.2, and compare this with the available information models, then we have to conclude that none of the information models fulfil all requirements. The first three information models summarised above are aimed at describing diagnostics information. NetConf and CIM are also aimed at the configuration management. These three information models are therefore aimed to inform the direct operators of those machines, and the topology part of the models are aimed at that context. The models only describe neighbour information, which makes it hard to describe a complete network topology with it.

The GMPLS information model is aimed at a very different public, namely the switches and routers themselves. The data model is aimed at compactness and is therefore not easy for other applications to understand, nor is it human readable. Unlike the first three information models, GMPLS is aimed at purely describing network topologies, so we take the information model of GMPLS into account, see also section 3.4.

The NM-WG and cNIS information models are aimed at both human and computer consumption. The goal of the models is to publish and share network measurement data along with topology data of those measurements. These measurements are stored in a database so that historical performance data is preserved. The main aim of the NM-WG data model is to relate it to the performance data. So it is not directly suited to create distributed domain descriptions. Nonetheless, the NM-WG model is a very good fit to our problem and is one of the important base models in the new NML-WG.

2.4 Topology Descriptions in Routing Protocols

In the previous section we have examined protocols and information models for describing computer networks. There are also other approaches to routing traffic and circuits in networks which do not require a complete overview of the network topology. For example routing in the backbone of the Internet is managed using the Border Gateway Protocol (BGP)[29, 47], and call routing in circuit switched telephone networks is handled with SS7 (Signalling System #7)[47].

BGP is the routing protocol used in the core routers of the Internet. Networks attached to these routers are identified using Autonomous System (AS) numbers. The exchanged topology information is defined with AS-paths for IP network prefixes. Reachability information that is published in BGP often does not travel through the whole network. A specific path can be aggregated together with other prefixes, saving space in the routing table. Another option is that prefix announcements are filtered out, when the same prefix can be reached through another way, or because of policy reasons. The limited topology information is very suitable for packet-switched networks, but not directly applicable to circuit-switched networks.

There has been an effort to attempt to combine optical networking with BGP called Optical BGP (OBGP)[48]. It uses virtual routers to exchange reachability information on behalf of optical cross connects (OXC). There are some difficulties with these approaches, most users of current lightpaths do not use public IP space, they are often on separate networks, which are temporarily connected to other resources. This makes it hard to announce the connectivity with prefixes. Another difficulty is that the technology used in the optical connectivity is relevant for the whole path construction, and it is not possible to encode this information in the OBGP announcements.

A completely different approach can be seen in telephony networks. Historically these circuit-switched networks have worked without or with very limited topology information. SS7 is the routing protocol currently used in telephony networks. Routing is performed in a distributed hierarchical fashion, based on dynamic lookups and default routes. This approach is feasible in telephone networks, because the network is provisioned and configured in such a way that blocking within the network is not very likely to occur. Such an approach is not applicable to optical networks, where blocking can easily occur.

2.5 Data Models

In section 2.2 we have shown that key requirements are interoperability and exchanging data between different parties. This makes it very important that we have a common, interoperable data model. All of the existing models we described in section 2.3, except the NM-WG model, are not suitable for distribution outside domains.

The information models used in routing protocols as described in section 2.4 are suitable for inter-domain distribution. However, these models have been specifically designed for the networks they are used in, and can not be applied without major modifications to optical networks.

The interoperability and suitability for exchange are mostly properties of the data model. Currently the lightpath provisioning is still done mostly by hand, so we also prefer a data model which is human readable or can be easily visualised, to help the transition towards full automation. There are many data models available, ranging from just plain text to complex binary serialized objects.

Most data models currently in use for computer networks and network management are aimed at information on a single layer. They are typically transported either by a dedicated protocol, or adapted to fit into header space of data packets. Needless to say, these data models are not very portable outside their subnets or protocols.

A standard data model that supports human-readability and interoperability is XML. Since a few years however, there is another data model gaining popularity: the Resource Description Framework (RDF). RDF has been developed by the W3C to support the idea of the Semantic Web.

We assume that the reader is familiar with XML[49, 50]. Below we provide a short introduction to the Semantic Web and RDF, before proceeding to compare RDF with XML.

2.5.1 Introduction to the Semantic Web

The World Wide Web has allowed us to publish and share documents and information with other people in the world. However, because the web is so popular and widespread, it has almost become the victim of its own success. Because of the large scale and the abundant availability of data, it becomes very hard to find what we want. Search-engines, such as Google or Yahoo, have come to the rescue and have indexed the data. However, computers still have no common sense, so the search capabilities of the search machines are rather limited. Consider for example the following two sentences:

- A is connected to B.
- There is a connection between A and B.

Even humans can differ in opinion whether these two sentences have the same meaning. So there is no way that a computer without common sense will understand that these two lines mean the same thing. This is where the Semantic Web comes to the aid of computers (and people). The following is an excerpt of the activity statement of the Semantic Web initiative [51]:

> *The goal of the Semantic Web initiative is to create a universal medium for the exchange of data where data can be shared and processed by automated tools as well as by people. For the Web to scale, tomorrow's programs must be able to share and process data even when these programs have been designed totally independently.*

In 2000 the Semantic Web initiative was started by the World Wide Web Consortium (W3C). Since then they have been working on several specifications to publish and share (meta)data, including the Resource Description Framework (RDF) [52] and SPARQL [53]. In the following section we provide a brief introduction to RDF.

2.5.2 Resource Description Framework

In order for two computer programs to communicate there must be a common understanding about the vocabulary being used. Currently most communication by computer programs is defined by protocols. The form of the interaction is fixed, but the meaning of the data being exchanged is not.

Take a web-browser for example: when a user types in a URL, the web-browser starts communicating with the designated server, asking for the resource identified by the URL ('GET'). The server then answers the browser with the designated file. This interaction is strictly defined in RFC 2616[54]. But neither the web-browser nor the server know what kind of data is being exchanged, it could be about the weather, traffic information, etc.

Because in this example the applications do not grasp the meaning of the data being presented, it limits the possibilities to mere presentation. The Semantic Web idea originated as a solution to this problem; it tries to make it easier for computers to understand the meaning of the content they present, so that they can navigate autonomously through this information to find what they are looking for.

The Resource Description Framework has been created to describe things in a meaningful way for computers. It provides a common framework for expressing metadata so that it can be exchanged between applications without loss of meaning.

Information in RDF is expressed as statements. Each statement is a triplet, with the following elements:

Subject The resource being described

Predicate The property of the subject that is described

Object The value of the property for the subject

A set of triplets is called a graph. An object can also be the subject of another triplet, so complex graphs can be created. An example of such a graph is shown in figure 2.1. An example triplet in the graph is '`Thesis creator Jeroen`', with the subject, predicate and object respectively. The graph also contains other triplets providing more information about the object `Jeroen`, such as his name, email address and homepage.

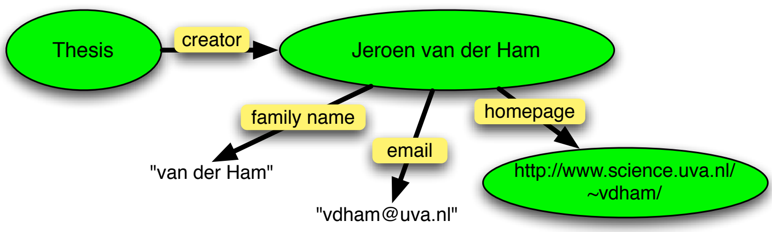

Figure 2.1: *A simple RDF graph*

The graph shown in figure 2.1 still has a problem; we have provided an abstract way of defining relations, but we still use plain English as labels for identifying these relations. Consider for example the `author` relationship, we could also have expressed this as `creator` without much loss of meaning to human readers. RDF solves this terminology problem by using Uniform Resource Identifiers (URIs)[2]. Related terms are usually defined using the same

[2] A URL is one kind of URI. See also [55].

URI-prefix, taking the form of XML namespaces. See for example the Dublin Core Metadata Initiative[56].

There are several ways of expressing RDF graphs, one is the graphical form as in figure 2.1. The most common textual form is *RDF/XML* [57], where the graph is encoded in an XML format. Throughout this thesis we use the RDF/XML notation, which allows us to leverage tools for XML as well as RDF.

2.5.3 RDF Schemata

An example of using semantics with data is the Friend of a Friend project[58]. Participants of this project describe themselves, giving their name, homepage, place of work, etc. The properties are predefined to make sure there is no conflict with e.g. using *'last name'* versus *'surname'*. But definitions of terms is not enough for computers: is *'surname'* the same as *'Surname'*? An example of a FOAF description is given below.

```
<foaf:Person rdf:nodeID="#me">
   <dc:creator rdf:resource="http://www.science.uva.nl/~vdham/thesis/">
   <foaf:family_name>van der Ham</foaf:family_name>
   <foaf:mbox>vdham@uva.nl</foaf:mbox>
   <foaf:homepage rdf:resource="http://www.science.uva.nl/~vdham/"/>
</foaf:Person>
```

Listing 2.1: *The RDF/XML representation of the semantic graph in figure 2.1*

Listing 2.1 describes the semantic graph of figure 2.1 in RDF/XML format. The properties in the example are defined using XML namespaces for readability, they actually point to specific URIs with a well-defined meaning. For example the `creator` property is defined by the URI `http://purl.org/dc/elements/1.1/creator`, which is defined by the the Dublin Core Initiative [59]. When defining RDF properties, it is possible to define what kind of types are valid as subject and object. The set of valid subjects is called the *domain*, and the set of valid objects is called the *range* of that property.

The other terms are either from the `rdf` namespace to describe standard RDF types and objects, or the `foaf` namespace, which provides definitions for the `Person` class, and basic properties of that class. Note that the `homepage` relationship points to a URL, but in this case it is also treated as an RDF object. This homepage object can then have other properties, such as a `creator` property.

An XML namespace with definitions of related terms is called an RDF schema. RDF schemata define the URIs and properties of RDF classes and RDF predicates. RDF classes define the types of subjects and objects.

```
1 <foaf:Person rdf:nodeID="#me">
2    <foaf:knows>
3       <foaf:Person>
4          <foaf:name>Freek Dijkstra</foaf:name>
5          <rdfs:seeAlso rdf:resource="http://staff.science.uva.nl/~fdijkstr/foaf.
               rdf"/>
6       </foaf:Person>
7    </foaf:knows>
8 </foaf:Person>
```

Listing 2.2: *Example of linking descriptions.*

2.5.4 Distributed Repositories

So far we have described how to create local descriptions. However, the Friend of a Friend project as described above is aimed at creating a distributed network of people descriptions. Each person publishes his own description, and links that with descriptions of other people. In RDF the `seeAlso` statement is used to create these pointers. See for example listing 2.2, which extends the earlier example.

Line 1 is the same as in listing 2.1, defining that the description of this person is about me. Line 2 defines that I have a `knows` relation with the `Person` object given in lines 3 to 6. In this case we do not provide an identifier of that object, but rather give a description of it. Line 5 points to another description file which contains more information about this object.

2.5.5 Comparing XML and RDF

Now that we have introduced RDF, we examine some differences between RDF and XML. The most important features in a multi-domain distributed descriptions context are the flexibility of the syntax, how object identifiers are created, how distributed descriptions can be handled, and extensibility.

XML stands for extensible markup language, and ordinary XML syntax is completely extensible, anything is allowed, as long as it is well-formed. This changes dramatically once schemata are defined for documents. These schemata are by definition restrictive.The schema must explicitly define places where extensibility is allowed. If external schemata are used for extensibility, then often an explicit version of that schema is chosen. XML schemata in itself do not contain versioning information. Currently, it is common practice to embed the version number in the name or URL of the schema. If an external schema is used to define some limited extensibility, then to use a new version, the original schema must also be updated.

RDF schemata take a different approach than XML schemata: rather than restrictively defining the structure of a (sub)document, RDF schemata define object classes and properties, and how they can be combined. This means that one document can use many schemata, and also that a single object can have properties defined in different schemata without depending on each other. This also means that when a new version of a certain schema is released, then only the documents need to be updated, since schemata tend not to point directly to other schemata.

The difference in the way schemata are defined also shows when we examine what kind of structures can be described using XML or RDF. In RDF an object is identified using its identifier, which makes it possible to define a flexible graph containing loops. XML on the other hand defines a tree structure, and uses that tree to define relations between objects. This means that it is not straightforward to describe a graph containing loops.

The identification of objects is also a strong feature of RDF. Objects must be identified using a URI, and it is common practice to use the URL of the document as part of that URI. This makes it very easy for authors to ensure global uniqueness of identifiers, which is an important aspect in a multi-domain environment. In XML there is no restriction on what kind of identifiers can be used, and it is not straightforward to create globally unique identifiers. It is also not possible for XML schemata to define restrictions on identifiers, so this must be defined externally.

Creating distributed descriptions is possible with both XML and RDF. In XML it is possible to link documents using XLink[60], or by defining links yourself. On the other hand in RDF the linking between descriptions is a built-in feature using `seeAlso` statements as described earlier. Unfortunately, there is little library support for either XLink or seeAlso, so parsers will have to implement this.

In terms of verbosity, XML has an advantage over the XML syntax of RDF. In our experience, RDF/XML is about twice as verbose as XML. It should be noted that RDF models can also be described using other more compact syntaxes, such as N3. These syntaxes are also supported by almost all RDF tools and libraries.

Another important difference is that RDF imposes some restrictions on the information model. When defining a schema for RDF, that is a mapping of the information model to the data model, a complete *ontology* has to be defined. This means that every element must be given a well-defined meaning. This is both an advantage to the schema author, who is forced to clearly define context and meaning for every single element, as well as for users, who may use the meaning to leverage the information on the semantic web.

2.6 Conclusion

In this chapter we have given an overview of the features of existing, well-known information models for describing computer networks. The first three information models we discussed, SNMP, NetConf and CIM, are aimed at diagnostics and configuration management. The information models of NM-WG, and G.805 are more aimed at topology description, however these models are not intended to publish topology information to other domains. The model of GMPLS is designed for use on the control plane between networking devices, and is not aimed at publishing to other domains, and is also not easily suitable for human consumption. We have to conclude that all these models are unsuitable to publish topology descriptions for inter-domain pathfinding in hybrid networks.

The most common data model is XML, which is suitable for distribution and human-readable. The other data models, for SNMP and OSPF, are specifically designed for use in their respective protocols, and these are not directly suitable for exchange outside of their protocols.

We have also examined another possible data model from the Semantic Web research, RDF. Data model may not be the proper label for RDF since it is more like a layer between the information model and data model. Data is described using triplets forming a graph, and these triplets can be encoded in several different data models, including XML.

In the previous section we have explained several advantages that RDF has over plain XML. We believe these features are important when developing a network schema:

- The use of URIs as generic identifiers is an advantage in multi-domain environments, since it makes it easy to express a request like 'a path from A to B' with A and B clearly and uniquely defined.
- We want to describe the interrelation of different (administrative) network domains. Each domain must be able to publish its own network information and point to other network domains. The seeAlso property makes it possible to easily link between different domain descriptions.
- We want to allow for easy extensibility of the network schema. That is, allow the users to not only publish network information they care about,

but also allow them to mix this with other schemata, both current (e.g. geographic information or organizational information in `geo` and `vcard`), and future schemata, either direct extensions of NDL or non-directly related schemata.

Because RDF/XML is an XML syntax, one could argue that XML can always be used to achieve all of the above strengths by defining an XML syntax that mimics the RDF syntax. However, in that case it is better to use an existing standard, than to develop a custom solution, both for compatibility reasons, as well as being able to leverage existing tools.